Combination of NN and CRF Models for Joint Detection of Punctuation and Disfluencies

Eunah Cho, Kevin Kilgour, Jan Niehues, Alex Waibel

Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{eunah.cho|kevin.kilgour|jan.niehues|alex.waibel}@kit.edu

Abstract

Inserting proper punctuation marks and deleting speech disfluencies are two of the most essential tasks in spoken language processing. This challenging task has prompted extensive research using various techniques, such as conditional random fields. Neural networks, however, are relatively under-explored for this task.

Combining different modeling techniques with different advantages has the potential to lead to improvements. In this work, we first establish the performance of joint modeling of punctuation prediction and disfluency detection using neural networks. We then combine a conditional random fields based model and a neural networks based model log-linearly, and show that the combined approach outperforms both individual models, by 2.7% and 3.5% in F-score for speech disfluency and punctuation detection, respectively. When used as a preprocessing step to machine translation this also results in an improved translation quality of 2.5 BLEU points compared to the baseline and of 0.6 BLEU points compared to the non-combined model.

Index Terms: speech disfluency detection, punctuation insertion, speech translation

1. Introduction

Processing of spontaneous language poses a great number of challenges in natural language processing (NLP) tasks, due to its distinctive characteristics compared to written language. While written language generally consists of well-formed, grammatically correct sentences, spontaneous speech very often contains disfluencies. Also, unlike text written by humans, conventional automatic speech recognition (ASR) systems do not provide reliable sentence boundaries and proper punctuation marks in their outputs.

These differences can negatively impact the performance of subsequent applications, such as machine translation (MT) systems. Most conventional MT systems are trained using written texts. When we deploy MT systems for spoken language, there is a mismatch between the training data and the output of the ASR system which is recognizing the spontaneous speech. As well as degrading the translation quality [1, 2], speech disfluencies and the lack of proper punctuation marks, greatly reduce the readability when presenting the recognition of spontaneous speech to users.

Conditional random fields (CRF) and neural networks (NN) have been used extensively for various NLP tasks, showing different advantages. CRFs are successfully used in sequence labeling tasks, due to their ability to model first order dependencies. NNs, on the other hand, have proven themselves to be very useful at classification tasks such as character recognition [3] and are therefore a sound choice for NLP tasks of this nature. The different strengths and weaknesses of both CRF models and NN models suggest that they can complement each other when jointly applied to the task of detecting punctuation and disfluencies in spontaneous speech. Despite the potential advantages they can offer when applied together, combining the two modeling techniques for punctuation and disfluency detection has not been investigated yet.

In this work, we present a punctuation and disfluency detection scheme using a combination of both CRF and NN models. We propose an NN designed to exploit the above mentioned synergistic effects by jointly modeling both punctuation and disfluencies in a single network with multiple parallel output layers. One output layer is devoted to detecting speech disfluencies while the other output layer is concerned with predicting punctuation marks. The CRF also models punctuation and disfluency detection using two output labels, where the first label covers disfluency and the second one punctuation marks. The predictions of the models are extracted in probabilities and used as features in a log-linear combination.

2. Related Work

In [4] the authors investigated three different methodologies of inserting punctuation marks within a given sentence boundary on TED talks[1] using a monolingual translation system. This is later extended in [5] so that the monolingual translation system can also detect sentence boundaries in lecture data. Their results demonstrated that inserting punctuation marks using another MT system is a promising way to improve the MT performance of ASR output. Fitzgerald et al. [6] applied a CRF-based disfluency detection model using lexical, language model (LM), and parser features.

In [7] the authors presented an extensive study on various methods of combining punctuation prediction and disfluency removal for telephone speech data. Their results demonstrate clearly that both problems influence each other. The soft cascade system, where the decision label of the first prediction is embedded as a feature of the second step, outperforms the hard cascade approach where the second step is only performed on the output of the first step.

The impact of segmentation and disfluency removal on translation of conversational speech is investigated in [8]. They separated the process into several steps. First they use a CRF model to detect sentence units. Based on these units they detect speech disfluencies, which are divided into two categories. After the simple disfluency is modeled using a CRF model, they use another CRF classifier to insert punctuation marks followed

[1] http://www.ted.com

by a knowledge-based parser in order to remove more complicated disfluencies.

Recently authors in [9] applied different segmentation and punctuation insertion schemes along with CRF-based disfluency detection on meeting data and translated the output in order to evaluate the performance in terms of BLEU [10]. In one system, they jointly modeled punctuation and disfluency using a CRF model, which improved the baseline by around 2 BLEU points. This performance, however, is around 0.5 BLEU points lower than their best system, where speech disfluency is modeled using a CRF model and punctuation mark insertion is modeled separately using their monolingual translation system. They attribute the performance drop to the limited amount of data for CRF modeling. Only disfluency annotated meeting data was used for the joint training while the monolingual translation system was trained on a larger data set.

3. Data

We chose to conduct our experiments on a set of transcribed English meetings and lectures introduced in [9]. Speech disfluencies are annotated with disfluency labels by human-annotators and grouped into five disfluency classes. The first class, `FL`, represents filler words, such as *"uh"* or *"uhm"*, and discourse markers, e.g. *"you know"* and *"like"*. The next class, `RC`, covers repetitions which can be either exact copies of utterances or simply a rough copy of them. In spontaneous speech a speaker may change her mind on what she wants to say or how she wants to express herself, leading to unfinished or partial sentences. These corrected parts are assigned the class `NC`. The last disfluency class, `IR`, is used for interruptions. In conversational speech, especially in multi-party meetings, there is a great deal of communication between different speakers. When a speaker is interrupted she may not be able to finish her sentence causing partial segments.

Words are grouped into four classes, `Period`, `Comma`, `QuMark` and `none` based on punctuation marks directly following them. The punctuation class `Period` is used for periods and exclamation marks. The class `Comma` only contains commas, while the class `QuMark` is used for question marks. Detailed statistics of the data is shown in Table 1. The class `none` represents the token which is not followed by a punctuation mark.

The training data set consists of both lectures and meeting data while the validation and test sets are made up of only meeting data. Compared to the lecture data the sentences in the meeting data tend to be shorter. They therefore on average contain fewer commas per sentence but more sentence boundary punctuation marks. The meeting data uses 5 meeting sessions with 5 to 12 speakers per session and the lecture data consists of 26 lectures each given by a single lecturer. The test data is manually translated into French, and used to evaluate the performance and effect of punctuation prediction and disfluency detection in MT.

		Train	Valid	Test
All tokens		142,789	8,466	14,855
Disf.	FL	7,447	674	999
	RC	4,743	450	1,017
	NC	2,634	146	330
	IR	1,361	250	860
	clean	126,604	6,946	11,649
Punct.	Period	7,809	485	1,621
	Comma	10,181	388	963
	QuMark	905	117	190
	none	123,894	7,476	12,081

Table 1: Data statistics.

Apart from the annotated data, we use 400K words of unannotated meeting data for pretraining the NN models, which we describe in 4.2. The un-annotated data has a format similar to that of the meeting data part of the training data.

4. Model

4.1. Features

The same set of features is used for both CRF and NN training in order to make them comparable. Inspired by [11], we use lexical and LM features. Our lexical features include the current and adjacent words/part-of-speech (POS) tokens in a 20 word window and their patterns. For NN training, we use the vector representation of each word as in [12]. Using the k-means algorithm words are also clustered into 100 clusters based on their vector value. Analogue to the features introduced in [2], we use the phrase table from an MT system to check the potential translations of each word. The LM features contain unigram and 4-gram scores as well as their ratios. Information about segment boundaries induced by speaker changes is also included as an additional feature.

4.2. Conditional Random Fields

As a framework dedicated to labeling sequence data, CRFs show good performance in diverse tasks of NLP, including sentence segmentation [13], POS tagging [14] and shallow parsing [15]. Our disfluency detection and punctuation insertion model can be also modeled as a sequence labeling task, where each word is labeled into different disfluency and punctuation classes.

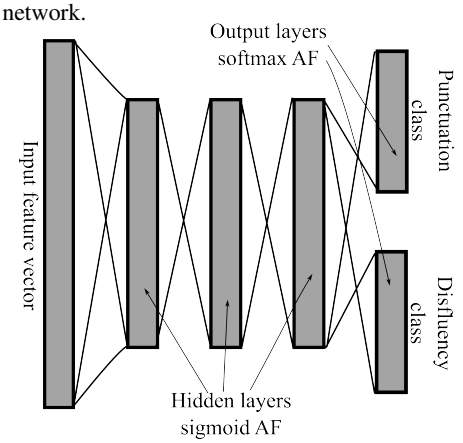
In this work we use the linear chain CRF modeling technique implemented in GRMM package [16]. As there are two output labels for each token, one for disfluency and another for punctuation, we use one linear chain edge across disfluency labels, another one across punctuation labels, and another for the in-between edges. The model is trained using L-BFGS, with default parameters.

4.3. Neural Networks

Many other NLP problems like language modeling [17] have been successfully addressed using NN due to their good classification ability. Two broad categories of NN are used for sequence modeling, recurrent neural networks [18, 19] where a hidden layer depends on the previous token's hidden layer and the feature of the current token to predict its label and feed forward neural networks [20, 21] which have a fixed input context.

Due to its shorter training time we use a five layer feedforward NN in this paper. It is trained to jointly predict both the punctuation and disfluency labels. As can be seen in Figure 1 the input consists of a 907 dimensional feature vector encoding the features described in Section 4.1, followed by three hidden layers containing 500 neurons each, and two parallel output layers. The hidden layers use the sigmoid activation function and the output layers use the softmax activation function. The parallel output layers are devised for the joint detection of disfluency and punctuation marks. Each output layer is considered to be a separate softmax group which results in the network generating

Figure 1: Proposed joint punctuation and disfluency prediction neural network.



a separate probability distribution for punctuation and disfluency labels.

The network is pretrained layer-wise using denoising auto-encoders [22] which enable us to also make use of the 400K unannotated examples as well as the 140k labeled examples. After pretraining the network is fine-tuned using mini-batch gradiant decent. Pretraining and fine-tuning were implemented using *Theano* [23].

5. Log-linear Combination

In MT, combining different translation models and LMs log-linearly in a decoder can greatly improve the translation quality [24]. The individual models are encoded as separate features and weighted using interpolation coefficients optimized on a validation set. Inspired by this, we combine our CRF based punctuation and disfluency prediction with our NN based one log-linearly using a label LM. For this task, we perform the search for the best label sequence as well as the optimization of the log-linear weights using an MT decoder [25].

For a given word sequence $w = w_1 \dots w_n$ we wish to find the best sequence of punctuation and disfluency labels $(p, d) = (p_1, d_1) \dots (p_n, d_n)$:

$$\underset{(p,d)}{\operatorname{argmax}} \sum_{i=1}^{m} \lambda_i \cdot f_i(w, p, d) \tag{1}$$

where m is the number of features and

$$p_i \in \{\texttt{Period}, \texttt{Comma}, \texttt{QuMark}, \texttt{none}\}$$
$$d_i \in \{\texttt{FL}, \texttt{RC}, \texttt{NC}, \texttt{IR}, \texttt{clean}\}.$$

We define input features from the two models ($M \in \{CRF, NN\}$) for each of the punctuation labels by:

$$f_{\hat{p}}^{M} = \sum_{j=1}^{n} \delta_{\hat{p},p_j} \cdot \log P_M(p_j|w) \tag{2}$$

and for each of the disfluency labels by:

$$f_{\hat{d}}^{M} = \sum_{j=1}^{n} \delta_{\hat{d},d_j} \cdot \log P_M(d_j|w) \tag{3}$$

The final input feature is derived from a 9-gram LM trained on the output labels of the training data. The LM is built using the SRILM Toolkit [26].

This formulation of the problem not only allows us to find the optimal label sequence, it can also be easily extended to incorporate further models.

6. Experimental Setup

6.1. MT System Description

The English-to-French translation system is built on 2.3 million parallel sentences. The training data includes written-style data such as the European Parliament data and the News Corpus data. It also includes a spoken-style data such as TED, which is used as in-domain data on which the models are adapted. Manual transcripts of some of the TED data are used as development data for the MT system.

We use a 4-gram LM built with the SRILM Toolkit as well as a bilingual LM [27] in order to extend source word context. For the optimization, we use the minimum error rate training (MERT) [28] in the phrase-based decoder [25]. The translation results are reported using case-insensitive BLEU.

6.2. Results

Our first experiment measures the quality of disfluency detection and punctuation insertion using precision, recall and the standard F-score metric. The scores presented in Table 2 measure whether a word was labeled as one of the disfluency classes or not. The results of the individual CRF and NN models, which are found in the first two rows of the table, show that the CRF model detects more disfluencies and therefore has a better recall performance. On the other hand, the NN model outperforms it on precision leading to fewer false detections. Their log-linear combination improves the F-score by 2.7% and seems to strike a balance between precision and recall.

System	F-score	Precision	Recall
CRF	53.90	68.83	44.29
NN	49.31	81.08	35.43
Log-lin. comb.	56.56	72.77	46.26

Table 2: Overview of the disfluency removal F-scores for our systems.

Similarly, the evaluation of our models' punctuation prediction capabilities show that while the NN model is the most precise at detecting punctuation marks, it is more conservative, and therefore has a lower recall than the CRF model. As can be seen in Table 3, we achieve our best performance on both F-score and recall when the models are combined. Both metrics are noticeably improved by the combination, F-score by 3.5% and recall by 5.5%.

System	F-score	Precision	Recall
CRF	58.22	60.23	56.34
NN	52.82	65.31	44.35
Log-lin. comb.	61.76	61.64	61.87

Table 3: Comparison of our CRF, NN and combined systems for punctuation prediction measured in F-score.

In order to evaluate not only the raw detection accuracy, but also its impact on an MT system, we use the punctuation-predicted, disfluency-removed test data as input data for the MT system described in Section 6.1. It is evaluated against a human translation of the oracle text where all annotated disfluent

words were removed and reference punctuation marks are inserted. Table 4 shows the results. In order to ensure a fair comparison, we use consistent segments for translation and evaluation in all tests; they span all tokens between speaker changes. In the baseline system, all disfluent words are kept and only just prior to the speaker change is a single sentence ending period inserted. The test data generated by our systems and the reference may also contain punctuation marks within these segments. A trivial rule-based disfluency removal system that only removes simple filler words such as *uh* or *uhm* is also listed in order to demonstrate the additional capabilities of our models.

System	BLEU
Baseline	14.42
+ No *uh*	14.94
CRF	16.32
NN	16.18
Log-lin. comb.	16.93
Oracle	22.76

Table 4: Translation scores after disfluency removal and punctuation insertion using various systems, measured in BLEU.

Removing disfluent words and inserting punctuation marks using only the CRF model improves the translation quality of the baseline system by 1.90 BLEU points. This approach also compares favorably to our trivial system, beating it by around 1.38 BLEU points. The NN model outperforms the trivial rule-based system by 1.24 BLEU points. Using both models in a decoder results in our best score of 16.93 BLEU, which beats the baseline system by 2.51 BLEU points and the CRF-based system by 0.61 BLEU points.

6.3. Analysis

The synergistic effect of the log-linear combination is presented in Table 5. The raw input contains a repetition, marked in bold letters, and is missing proper punctuation marks. In the manually cleaned version of this excerpt, the repeated part is removed and punctuation marks are inserted, which makes it notably easier to understand. The CRF model was able to successfully detect the repetition in the first segment. In the second segment however, it deletes too much, leading to the ungrammatical sentence *"for what are these recordings for"*. This false labeling of disfluencies is probably due to the repetitive nature of that segment. The CRF also fails to insert any sentence boundaries. Although the NN based model was unable to remove the repetitive part in the first segment, it correctly detected the sentence boundary after the first segment.

Using the combined model we were able to remove the speech disfluency detected by the CRF model while at the same time inserting the correct sentence boundaries. This shows that even when the two separated models perform imperfectly, we can benefit from their synergistic effects. It is also notable that while the location of the sentence boundaries was correctly predicted by the NN it predicted the wrong punctuation class. In the model combination though both the location of the sentence boundaries and the fact that they were question marks were correct. It suggests that combining the models provides an opportunity to optimize on relative importance of the features.

Table 6 shows another impressive impact of the combined model on a very disfluent segment. Speech disfluencies according to the annotators are marked in bold letters. Although the result of the combined model does not match disfluency and punctuation marks of the annotation, thereby lowering the F-score, its readability is comparable to the annotated sentence.

Raw input	**do you use do you have** digits as a class what are these for what are these recordings for
Manually cleaned	Do you have digits as a class? What are these for? What are these recordings for?
CRF	do you have digits as a class *for what are these recordings for*
NN	**do you use do you have** digits as a class. What are these for what these recordings for
Log-lin. comb.	do you have digits as a class? What are these for? What are these recordings for

Table 5: Excerpt from the test data showing how the CRF and NN models complement each other.

Raw input	yeah **you** you mean this okay **right right** good **yeah** it's an at sign
Manually cleaned	Yeah. You mean this. Okay. Good. It's an at sign.
CRF	yeah, this okay **right, right** good **yeah,** it's an at sign.
NN	yeah, you mean this okay, **right, right** good, **yeah,** it's an at sign.
Log-lin. comb.	yeah, this okay, **right?** Good, **yeah,** it's an at sign.

Table 6: Excerpt demonstrating the improved readability of the combined model despite a prediction that is very dissimilar to manually cleaned text.

7. Conclusion

In this paper we showed that multiple models with complementary advantages can be combined in order to improve the performance of joint disfluency and punctuation labeling. We present both CRF based and NN based models and explain how they can be combined in a log-linear decoder, in order to achieve better performance. Both models and their combination are tested on conventional meeting data and intrinsically evaluated with F-score as well as extrinsically by using them as a precursory step to an MT system.

The results demonstrate that our combination outperforms the two individual models on both F-score and BLEU. Compared to the best single model it boosts the disfluency detection F-score by 2.7% and the punctuation prediction F-score by 3.5%. While both the CRF and NN models improve the translation quality of the baseline system by 1.90 and 1.76 BLEU points respectively, the combined approach gives us an improvement of 2.51 BLEU points.

An analysis of our proposed model indicates that these improvements stem from synergies between the models. We go on to show that it can also noticeably increase the readability of the spoken language input even when the model's output does not conform to the human annotation.

In future work, we would like to include further models into our combination and apply it to other genres and to other languages.

8. References

[1] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence Segmentation and Punctuation Recovery for Spoken Language Translation," in *ICASSP*, Las Vegas, Nevada, USA, April 2008.

[2] E. Cho, T.-L. Ha, and A. Waibel, "CRF-based Disfluency Detection using Semantic Features for German to English Spoken Language Translation," in *IWSLT*, Heidelberg, Germany, 2013.

[3] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[4] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *IWSLT*, San Francisco, CA, USA, 2011.

[5] E. Cho, J. Niehues, and A. Waibel, "Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System," in *IWSLT*, Hong Kong, China, 2012.

[6] E. Fitzgerald, K. Hall, and F. Jelinek, "Reconstructing False Start Errors in Spontaneous Speech Text," in *EACL*, Athens, Greece, 2009.

[7] X. Wang, K. C. Sim, and H. T. Ng, "Combining Punctuation and Disfluency Prediction: An Empirical Study," in *Conference on Empirical Methods on Natural Language Processing (EMNLP 2014)*, Doha, Qatar, 2014.

[8] H. Hassan, L. Schwartz, D. Hakkani-Tür, and G. Tur, "Segmentation and Disfluency Removal for Conversational Speech Translation," in *Interspeech*, September 2014.

[9] E. Cho, J. Niehues, and A. Waibel, "Machine Translation of Multi-party Meetings: Segmentation and Disfluency Removal Strategies," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, USA, 2014.

[10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation." IBM Research Division, T. J. Watson Research Center, Tech. Rep. RC22176 (W0109-022), 2002.

[11] E. Fitzgerald, F. Jelinek, and R. Frank, "What Lies Beneath: Semantic and Syntactic Analysis of Manually Reconstructed Spontaneous Speech," in *ACL*, Singapore, 2009.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Workshop at ICLR*, Scottsdale, AZ, USA, 2013.

[13] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using Conditional Random Fields for Sentence Boundary Detection in Speech," in *ACL*, Ann Arbor, MI, USA, 2005.

[14] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilitic Models for Segmenting and Labeling Sequence Data," in *ICML*, Williamstown, MA, USA, 2001.

[15] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," in *HLT/NAACL*, Edmonton, Canada, 2003.

[16] C. Sutton, "GRMM: A Graphical Models Toolkit," 2006. [Online]. Available: http://mallet.cs.umass.edu

[17] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[18] T. Mikolov, M. Karafiat, J. Cernocky, and S. Khudanpur, "Recurrent Neural Network based Language Model," in *Interspeech*, 2010.

[19] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling." in *INTERSPEECH*, 2012.

[20] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *AISTATS*, 2005.

[21] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," in *Innovations in Machine Learning*. Springer, 2006, pp. 137–186.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[23] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a cpu and gpu math expression compiler," in *Proceedings of the Python for scientific computing conference (SciPy)*, vol. 4. Austin, TX, 2010, p. 3.

[24] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *ACL*, 2002.

[25] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[26] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *ICSLP*, Denver, CO, USA, 2002.

[27] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *WMT*, Edinburgh, UK, 2011.

[28] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *WPT*, Ann Arbor, MI, USA, 2005.